README

OrgSync

Built by Hank Hayes for CS392E: Mobile Computing

Table of contents

About

OrgSync is a student organization management application. It aims to give student organizations an easy way to manage their members, send announcements, and plan events.

Dependencies

- Swift version: `Apple Swift Version 5.9`
- Xcode version: `15.0.1 (15A507)`
- iOS Deployment target: `17.0`

Packages

- Firebase: `https://github.com/firebase/firebase-ios-sdk`

Special instructions

- Simulator: `iPhone 15 Pro Max`
- Orientation: `Portrait`
- Test account email: `hankhayes@utexas.edu`
- Test account password: `adminone`

Sample data for members, events, and announcements have been already created. There's no need to launch multiple simulators, but if you'd like to see the Firebase Realtime Database in action, this would be the best way to do so.

> Registering as a new user will give you "member" privileges, so you will not be able to do things like create announcements or events.

Usage

OrgSync is built on top of a FireBase Realtime Database, which allows students to create their own membership account. Announcements and events that are created in the app are actually viewable on other devices at the time of creation.

> There are a few parts of this app that I started building and have not completed. All core project requirements are fulfilled but you may encounter an occasional bug or unfinished section.

Project requirements

General

☑ ~~Settings screen.~~

The implemented behaviors are **dark mode, haptics, and alternate app icons**.

- `SettingsTVC.swift`
- `SettingsViewController.swift`

☑ ~~Non-default fonts and colors used~~

- Verdana & System Indigo

Two major elements used

☑ ~~Login/register path with Firebase~~

- `WelcomeViewController.swift`
- `LoginViewController.swift`
- `SignUpViewController.swift`

□ Core Data

□ User profile path using camera and photo library

☑ ~~Multithreading~~

- `EventsViewController.swift` refreshing table views

□ SwiftUI

Minor elements used

☑ ~~Two or more additional view types~~

Segmented controllers, picker views, switches, and menu buttons

- `SettingsTVC.swift` changes settings using segmented controllers and switches
- `AnnouncementViewController.swift` has a menu button that allows filtering and announcement creation
- `CreateAnnouncementViewController.swift` uses a picker view to choose an announcement tag

At least one of the following

☑ ~~Table View~~

- `MemberViewController.swift` displays members in a UITableView

□ Collection View

☑ ~~Tab Bar View Controller~~

- `TabBarController.swift` controls navigation throughout the app

□ Page View Controller

At least one of the following

☑ ~~Alerts~~

- `AnnouncementViewController.swift` serves an alert upon successfully creating an event

□ Popovers

□ Stack Views

□ Scroll Views

☑ ~~Haptics~~

- `TabBarController.swift` plays a haptic when the tab is changed

☑ ~~User Defaults~~

- `SettingsTVC.swift` stores user settings in User Defaults

At least one of the following

□ Local Notifications

□ Core Graphics

□ Gesture Recognition

□ Animation

☑ ~~Calendar~~

- `EventDetailViewController.swift` lets you add an event to your calendar

□ Core Motion

□ Core Location / MapKit

□ Core Audio

□ Firebase (if not used to fulfill major element requirement)

□ Core Data (if not used to fulfill major element requirement)

□ Other

Contact

hankhayes@utexas.edu